확률 모델링

모듈 - 1

강사: 장순용 박사

순서

1. 확률 모델링:

언어 모형

언어 모형 (language Model)에 대해서:

- 단어 시퀀스의 확률을 예측하려 한다: $P(w_1, w_2, w_3, \ldots, w_i)$

  주의: w의 서브 인덱스는 순서대로 정렬되어 있다.

- 다음과 같이 단어 시퀀스가 주어질 때 $\{w_1, w_2, w_3, \ldots, w_{i-1}\}$, 이후 w_i로 이어질 확률은?

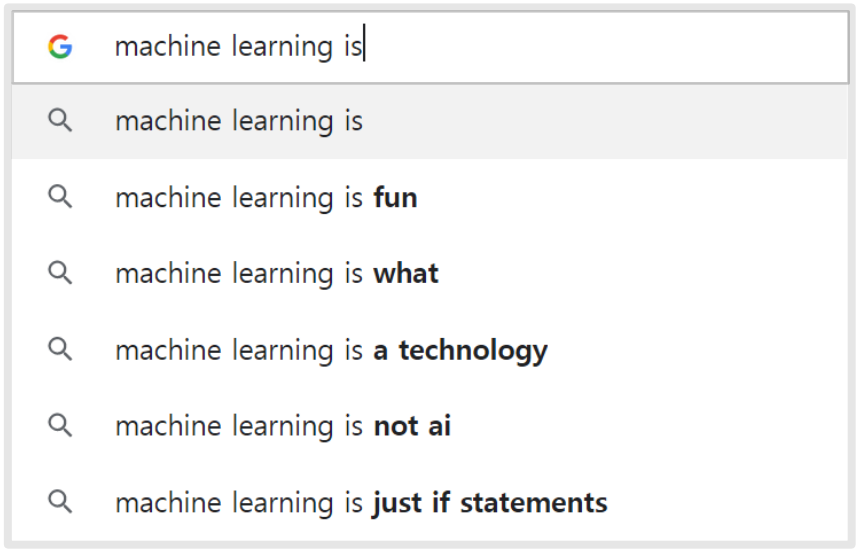
$$P(w_i|w_1, w_2, w_3, \ldots, w_{i-1})\ ?$$

- 데이터의 부족 (sparsity)은 큰 문제이다. 길고 같은 문장이 반복될 확률은 매우 낮다.
- 활용 분야: 자동 번역, 음성 인식, 철자법 확인, auto fill, 등.

언어 모형

언어 모형 (language Model)에 대해서:

예). 검색 창:



확률의 체인 룰 (Probability Chain Rule):

- 결합 확률을 다음과 같이 조건부 확률로 전개할 수 있다:

$$P(w_1, w_2, w_3, \ldots, w_m) = P(w_1)P(w_2|w_1)P(w_3|w_1, w_2)P(w_4|w_1, w_2, w_3) \cdots P(w_m|w_1, w_2, \ldots, w_{m-1})$$

$$= \prod_{i=1}^{m} P(w_i|w_1, \ldots, w_{i-1})$$

예). $P(three\ little\ pigs\ lived\ happily)$?

$\Rightarrow$

w_1	w_2	w_3	w_4	w_5
"three"	"little"	"pigs"	"lived"	"happily"

$$P(three\ little\ pigs\ lived\ happily)$$

$$= P(three)P(little|three)P(pigs|three\ little)P(lived|three\ little\ pigs)P(happily|three\ little\ pigs\ lived)$$

언어 모형

n-Gram에 대해서:

- 문장이 주어지면, 길이가 n인 "Moving Window"로 훑어 가면서 n-gram을 만들어 갈 수 있다.

예). "three little pigs lived happily"

→ $n = 1$, Unigrams = ["three", "little", "pigs', "lived", "happily"]

→ $n = 2$, Bigrams = ["three little", "little pigs", "pigs lived", "lived happily"]

→ $n = 3$, Trigrams = ["three little pigs", "little pigs lived", "pigs lived happily"]

"three little pigs lived happily"

"three little pigs lived happily"

"three little pigs lived happily"

언어 모형

n-Gram으로 근사:

- 문장이 길어질 수록, 데이터 부족으로 확률을 계산하는 것이 어려워 진다:

$$P(w_i|w_1, w_2, w_3, \dots, w_{i-1}) = \frac{Count(w_1, w_2, w_3, \dots, w_i)}{Count(w_1, w_2, w_3, \dots, w_{i-1})}$$

- 완전히 정확한 계산을 하기 보다는 다음과 같이 n-Gram으로 근사할 수 있다:

$$P(w_1, w_2, w_3, \dots, w_m) \approx \prod_{i=1}^{m} P(w_i|w_{i-(n-1)}, \dots, w_{i-1})$$

  $\Rightarrow$ 위 근사식을 아래 정확한 표현식과 비교해 볼 수 있다:

$$P(w_1, w_2, w_3, \dots, w_m) = \prod_{i=1}^{m} P(w_i|w_1, \dots, w_{i-1})$$

- 보통 n은 작은 양의 정수이다 $\cong 1, 2, 3, \dots$

언어 모형

n-Gram으로 근사:

- 만약에 $n = 1$이라면 Unigram 근사가 된다:

$$P(w_1, w_2, w_3, \dots, w_m) \approx P(w_1)P(w_2)P(w_3)\cdots P(w_m)$$

- 만약에 $n = 2$이라면 Bigram 근사가 된다 (마르코프 연쇄):

$$P(w_1, w_2, w_3, \dots, w_m) \approx P(w_1)P(w_2|w_1)P(w_3|w_2)\cdots P(w_m|w_{m-1})$$

- 만약에 $n = 3$이라면 Trigram 근사가 된다:

$$P(w_1, w_2, w_3, \dots, w_m) \approx P(w_1)P(w_2|w_1)P(w_3|w_2, w_1)P(w_4|w_3, w_2)\cdots P(w_m|w_{m-1}, w_{m-2})$$

예). Sequence = "*three little pigs lived happily* "를 Bigram 근사로 표현하면,

$$P(Sequence) \approx P(three)P(little|three)P(pigs|little)P(lived|pigs)P(hapilly|lived)$$

실습 #0101

→ 사용: **ex_0101.ipynb** ←

순서

1. 확률 모델링:

    1.1. 언어 모형.

    1.2. 나이브 베이즈 분류기.

    1.3. HMM 모델과 활용.

    1.4. 연관성 분석 (추천 시스템).

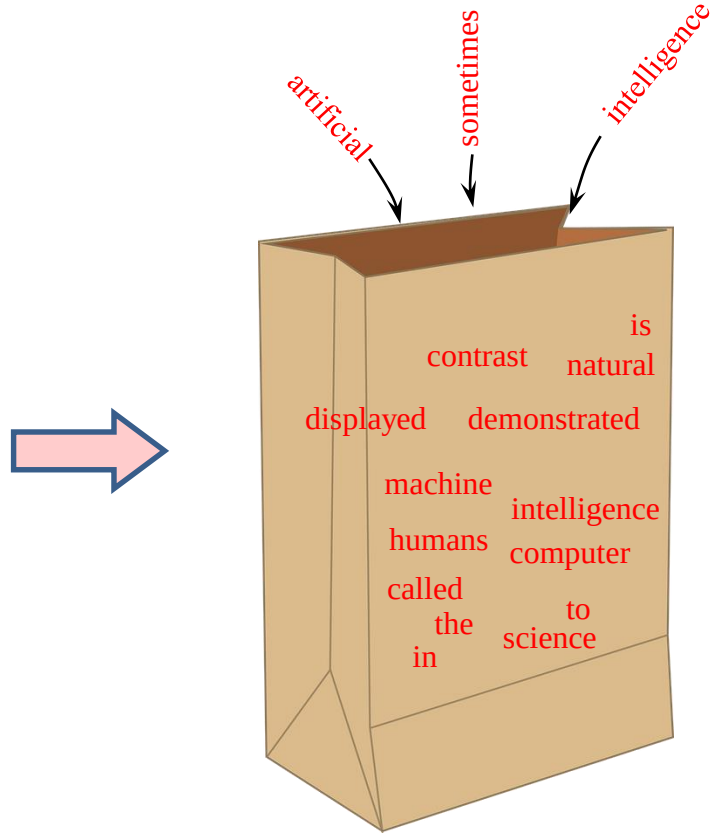
Naïve Bayes 분류기

Bag-of-Words (BOW) 자연어 모형:

- 문서는 단어(*)의 집합으로 이루어 졌다고 가정한다. (*) 또는 n-gram.
- 단어가 배열된 **순서나 문법은 무시**한다.
- 단어는 서로 독립적이며 단어의 **빈도수만 중요** 함.

예).

"In computer science, artificial intelligence, sometimes called machine intelligence, is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans."



Naïve Bayes 분류기

자연어 Naïve Bayes 분류기:

- 타입 A 와 타입 B 두 가지 유형의 문서가 있다고 전제한다.

  예를 들어서 A= “스팸 메일”, B = “스팸이 아닌 메일”.

- BOW 모형을 전제하며, 분절 된 단어를 담은 A 와 B 두 개의 bag가 있다고 가정한다.
- 베이즈 (Bayes) 통계법을 적용하면:

$$P(A|w_1, w_2, w_3, \dots) = \frac{P(w_1, w_2, w_3, \dots|A)P(A)}{P(w_1, w_2, w_3, \dots)}$$

$$P(B|w_1, w_2, w_3, \dots) = \frac{P(w_1, w_2, w_3, \dots|B)P(B)}{P(w_1, w_2, w_3, \dots)}$$

주의: 여기에서 w_i의 서브 인덱스 i에는 정렬의 의미가 없고 레이블링 용도만 있다.

Naïve Bayes 분류기

자연어 Naïve Bayes 분류기:

- 조건부 확률 $P(A|w_1, w_2, w_3, \dots)$와 $P(B|w_1, w_2, w_3, \dots)$를 비교해서 예측할 수 있다.
- 그런데, 비교에서는 상대적 크기가 중요하다.

  $\Rightarrow$ **물음**: 어느 쪽 조건부 확률이 더 큰가?
- 비교를 위해서 공통적인 분모 $P(w_1, w_2, w_3, \dots)$는 필요 없다.

$$P(A|w_1, w_2, w_3, \dots) \sim P(w_1, w_2, w_3, \dots|A)P(A)$$

$$P(B|w_1, w_2, w_3, \dots) \sim P(w_1, w_2, w_3, \dots|B)P(B)$$

- 단어가 독립적으로 발생했다는 전제를 하므로, 다음과 같은 전개가 가능하다.

$$P(A|w_1, w_2, w_3, \dots) \sim P(w_1|A)P(w_2|A)P(w_3|A) \cdots P(A)$$

$$P(B|w_1, w_2, w_3, \dots) \sim P(w_1|B)P(w_2|B)P(w_3|B) \cdots P(B)$$

Naïve Bayes 분류기

자연어 Naïve Bayes 분류기:

- 확률을 직접 비교하기 보다는 확률의 로그를 비교하기로 한다.

  ⇒ 작은 확률을 여러 번 거듭해서 곱하다 보면 컴퓨터로 표현할 수 있는 precision보다 작아질 수 있다.

- 이전 등식의 양쪽에 $Log()$ 함수를 적용하면:

$$Log(P(A|w_1, w_2, w_3, \dots)) \sim Log\big(P(w_1|A)\big) + Log\big(P(w_2|A)\big) + Log\big(P(w_3|A)\big) + \cdots + Log(P(A))$$

$$Log(P(B|w_1, w_2, w_3, \dots)) \sim Log\big(P(w_1|B)\big) + Log\big(P(w_2|B)\big) + Log\big(P(w_3|B)\big) + \cdots + Log(P(B))$$

⇐ 학습 데이터에서 A 문서의 개수 $=$ B 문서의 개수와 같이 맞추어 주었다는 전제가 포함되어 있다.

⇐ 즉, $Log(P(A)) = Log(P(B))$인 것이고 비교 목적에는 필요가 없으므로 제외 되었다.

Naïve Bayes 분류기

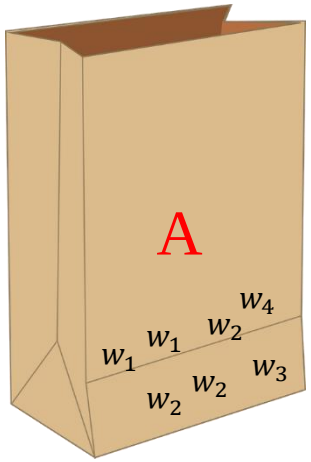
자연어 Naïve Bayes 분류기:

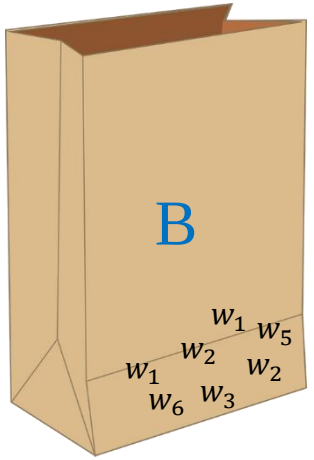
- 학습 단계:

1). A bag의 모든 단어에 대해서 확률 $P(w_i|A)$ 과 이것의 로그 $Log\big(P(w_i|A)\big)$를 계산해 둔다.

2). B bag의 모든 단어에 대해서 확률 $P(w_i|B)$ 과 이것의 로그 $Log\big(P(w_i|B)\big)$를 계산해 둔다.

3). 다음 예측을 위해서 위 스텝에서 계산해 둔 로그 확률을 저장해 둔다.





Naïve Bayes 분류기

자연어 Naïve Bayes 분류기:

- 예측 단계:

1). 새롭게 $w'_1, w'_2, w'_3, \ldots$ 와 같은 단어가 포함된 test 문서가 있다면 다음과 같이 두 가지 합을 구한다.

$$LogProbA = Log\big(P(w'_1|A)\big) + Log\big(P(w'_2|A)\big) + Log\big(P(w'_3|A)\big) + \cdots$$

$$LogProbB = Log\big(P(w'_1|B)\big) + Log\big(P(w'_2|B)\big) + Log\big(P(w'_3|B)\big) + \cdots$$

2). $LogProbA > LogProbB$ 이면: test 문서의 유형은 A라고 예측한다.

$LogProbA < LogProbB$ 이면: test 문서의 유형은 B라고 예측한다.

실습 #0102

→ 사용: **ex_0102a.ipynb , ex_0102b.ipynb** ←

순서

1. 확률 모델링:

    1.1. 언어 모형.

    1.2. 나이브 베이즈 분류기.

    1.3. HMM 모델과 활용.

    1.4. 연관성 분석 (추천 시스템).

은닉 마르코프 모델 : 개요

은닉 마르코프 모델 (Hidden Markov Model, HMM)에 대해서:

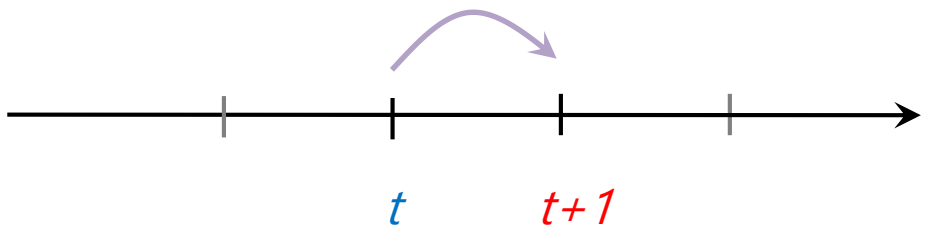
- 자연어 분석 (품사 태깅), 음성인식, 강화학습, 등 AI에서 많이 사용된다.
- 통신, 트레이딩 전략 등에도 사용된다.

은닉 마르코프 모델 : 개요

마르코프 과정에 대해서:

- 미래의 확률이 바로 한 스텝 이전과 연결되며 더 오래된 과거는 필요 없다.

$$P(x_{t+1}|x_t, x_{t-1}, x_{t-2}, \dots, x_1) = P(x_{t+1}|x_t)$$



- 마르코프 연쇄는 시간이 이산적인 경우에 해당한다.
- 실제 상태는 직접 관찰 가능하다.

은닉 마르코프 모델 : 개요

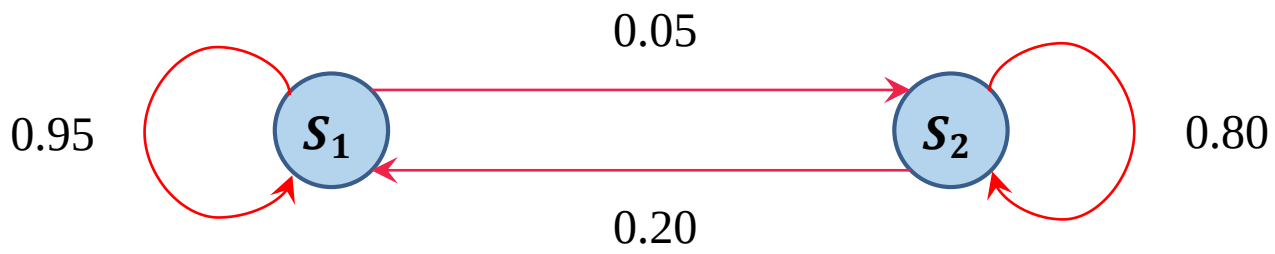
마르코프 과정의 예: #1.

- 다음과 같은 상황을 마르코프 과정으로 표현해 본다.

⇒ 오늘 A사의 주가가 상승했으면 내일도 95%의 확률로 주가가 오를것 이다.

⇒ 또한, 오늘 A사의 주가가 하락했더라도 내일은 20%의 확률로 주가가 오를 것이다.

⇒ 상승 상태는 S_1로 표기하고 하락 상태는 S_2로 표기한다.



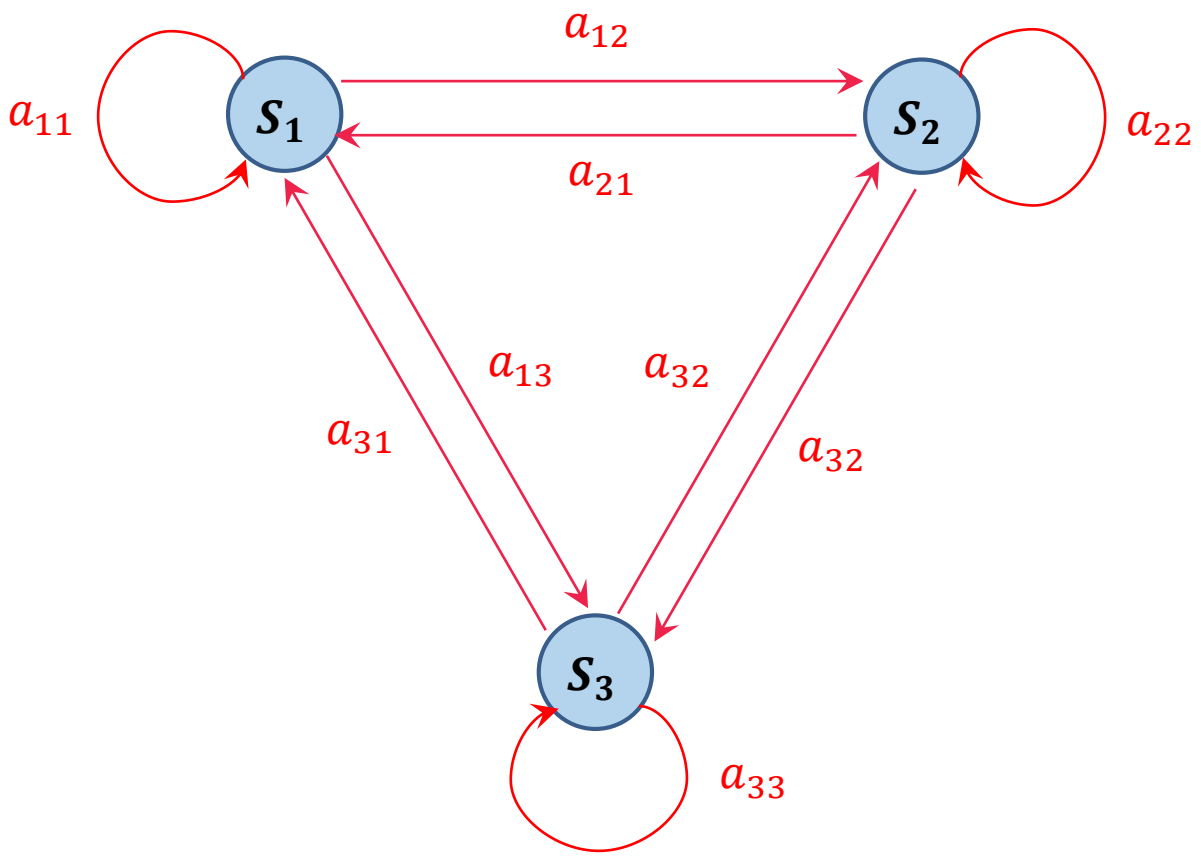
은닉 마르코프 모델 : 개요

마르코프 과정의 예: #2.

- 3 개의 관찰 가능한 상태 S_1, S_2, S_3이 있는 상황을 마르코프 과정으로 표현해 본다.

$\Rightarrow a_{ij}$ =전이확률.

$\Rightarrow \sum_j a_{ij} = 1$이다.



은닉 마르코프 모델 : 개요

마르코프 과정:

- 다음과 같이 전이해 갈 확률은?

t : 1 2 3

$$S_1 \rightarrow S_2 \rightarrow S_1$$

$\Rightarrow P(x_1 = S_1, x_2 = S_2, x_3 = S_1) = P(x_1 = S_1)P(x_2 = S_2|x_1 = S_1)P(x_3 = S_1|x_2 = S_2)$

$\Rightarrow$ 한 스텝씩 전진해 나가며 전이확률을 곱해준다.

- 일반화 해서 $x_1, x_2, x_3, \cdots, x_T$와 같은 시퀀스의 확률은 다음과 같다.

$$P(x_1, x_2, x_3, \cdots, x_T) = P(x_1)P(x_2|x_1) \cdots P(x_{T-1}|x_{T-2})P(x_T|x_{T-1})$$

- 언어 모형 (Language Model)의 Bigram 근사는 바로 마르코프 과정(연쇄)의 한 예이다:

$$P(w_1, w_2, w_3, \ldots, w_m) \approx P(w_1)P(w_2|w_1)P(w_3|w_2) \cdots P(w_m|w_{m-1})$$

은닉 마르코프 모델 : 개요

은닉 마르코프 모델이란?

- 다음과 같은 전제와 함께 은닉 마르코프 모델을 정의한다.

  $\Rightarrow$ 직접 관찰이 불가능한 (은닉된) 마르코프 과정(연쇄)를 따르는 상태 $S_1, S_2, \ldots, S_N$가 있다.

  $\Rightarrow$ 직접 관찰이 가능한 값 $O_1, O_2, \cdots, O_M$이 있다. $N = M$ 또는 $N \neq M$.

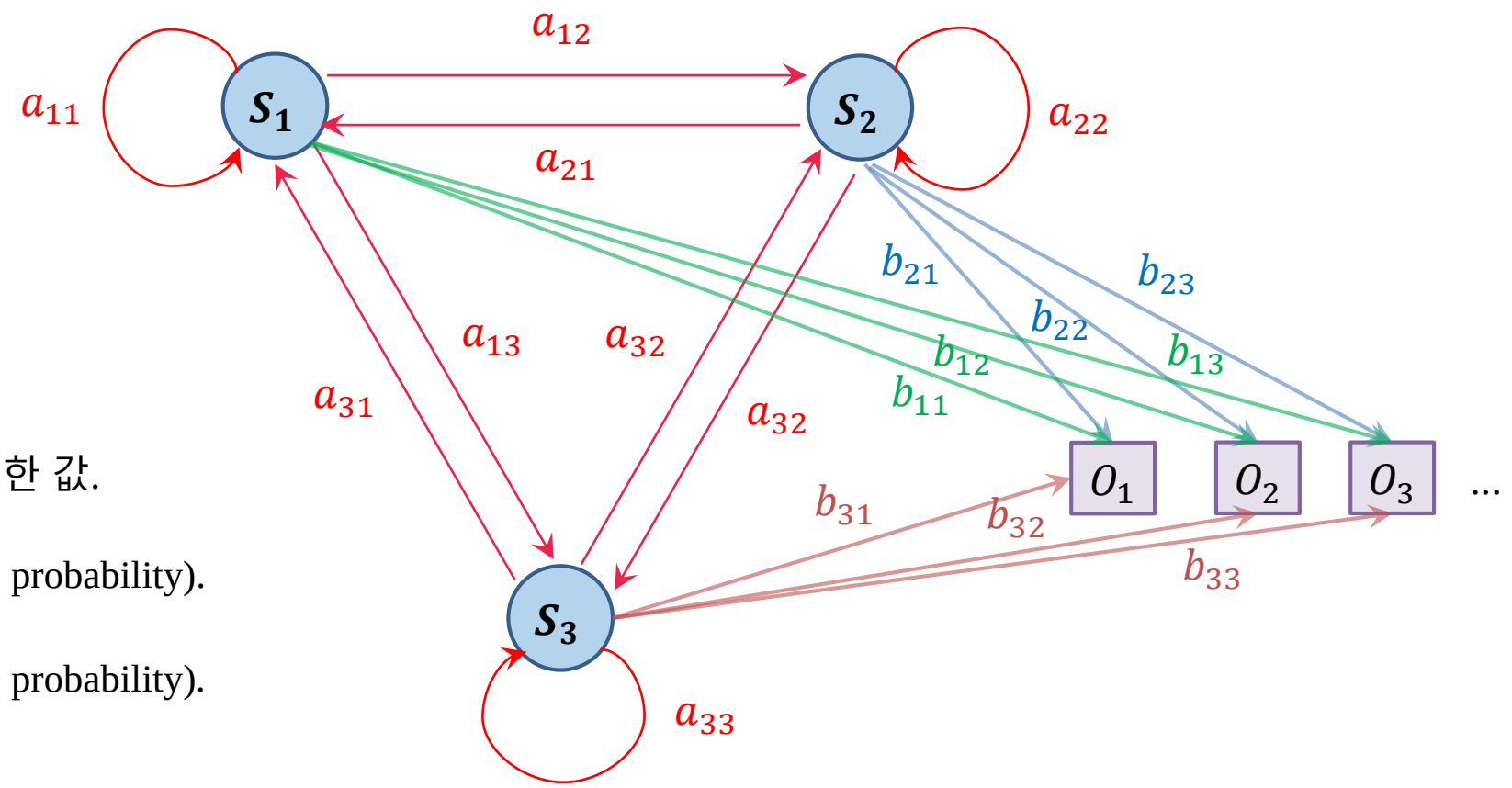
  $\Rightarrow S_i \rightarrow S_j$ 전이에 해당하는 확률은 a_{ij}이다. (transition probability)

  $\Rightarrow S_i$ 가 O_j로 관찰될 확률은 b_{ij}이다. (emission probability)

  $\Rightarrow$ 초기상태를 나타내는 확률분포 $p(x = S_i), 1 \leq i \leq N$가 있다.

은닉 마르코프 모델 : 개요

은닉 마르코프 모델의 예:



$\Rightarrow S_1, S_2, S_3 =$ 은닉된 상태.

$\Rightarrow O_1, O_2, O_3, \ldots =$ 관찰 가능한 값.

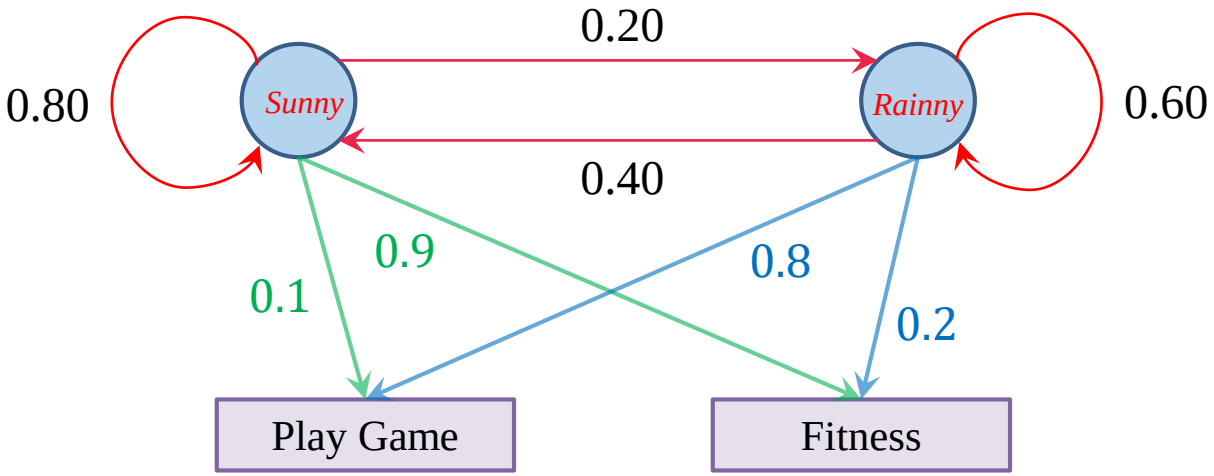
$\Rightarrow a_{ij} =$전이확률 (transition probability).

$\Rightarrow b_{ij} =$ 출력확률 (emission probability).

은닉 마르코프 모델 : 개요

은닉 마르코프 모델의 예: Wikipedia 참고

- 영희와 철수는 멀리 떨어져서 살고 있기 때문에 안부를 전화로 물을 수 밖에 없다. 철수의 일과는 크게 “게임 하기” 또는 “피트니스 하기” 두 가지로 있는데, 무엇을 할지는 그 날의 날씨에 따라 결정된다.
- 영희는 철수가 살고 있는 지역의 날씨에 관해서 정확히는 모르고 “확률적” 성향만을 알고 있을 뿐이다. 영희는 철수와의 통화내용에 기반하여 그 지역의 날씨를 예측해보려고 한다.

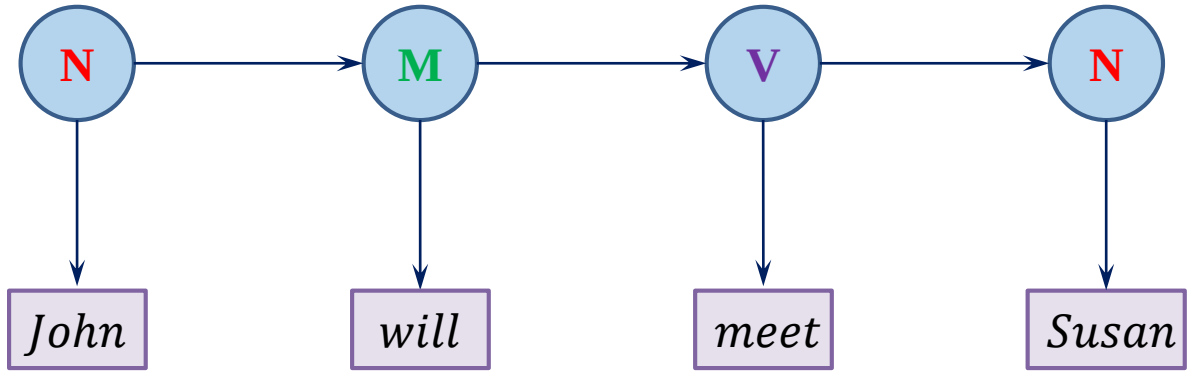


은닉 마르코프 모델 : 품사 태깅

품사 태깅 문제:

- 예를 들어서 “John will meet Susan” 이라는 문장이 관찰 되었다.
- 단어별로 품사를 태깅 하고자 한다.

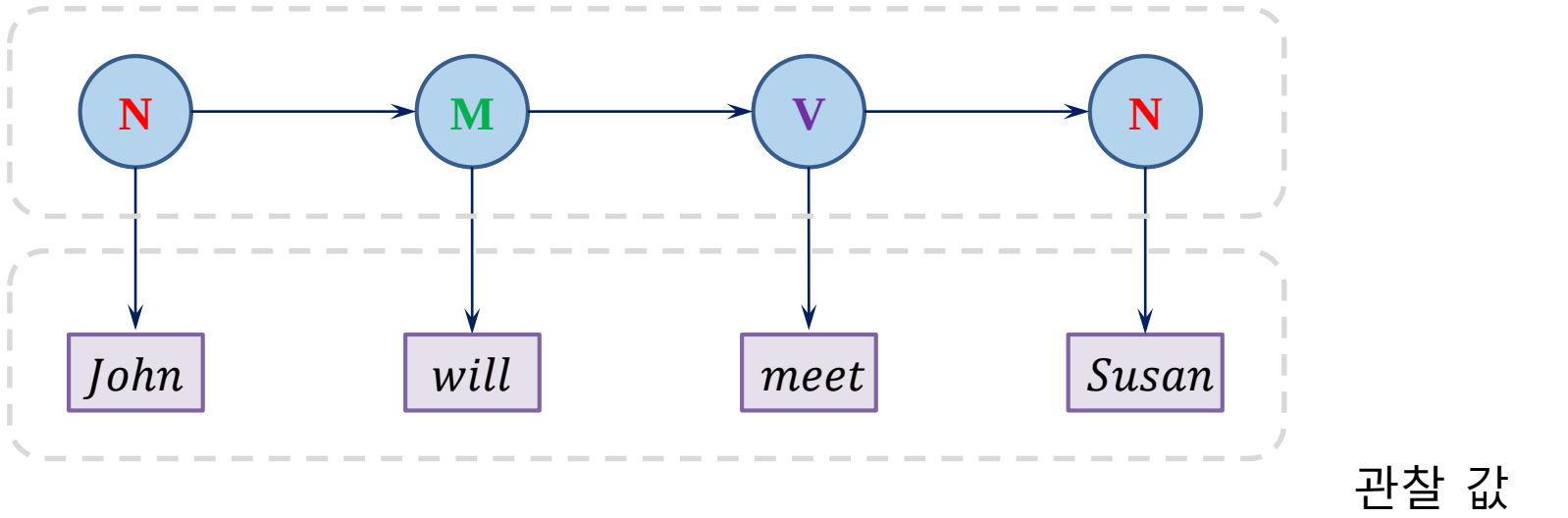
  ⇒ N=명사, M=조동사, V=동사.



은닉 마르코프 모델 : 품사 태깅

품사 태깅 문제:

- 은닉되어 있는 상태 → 품사 태그.
- 들어나 있고 관찰 가능한 값 → 단어.

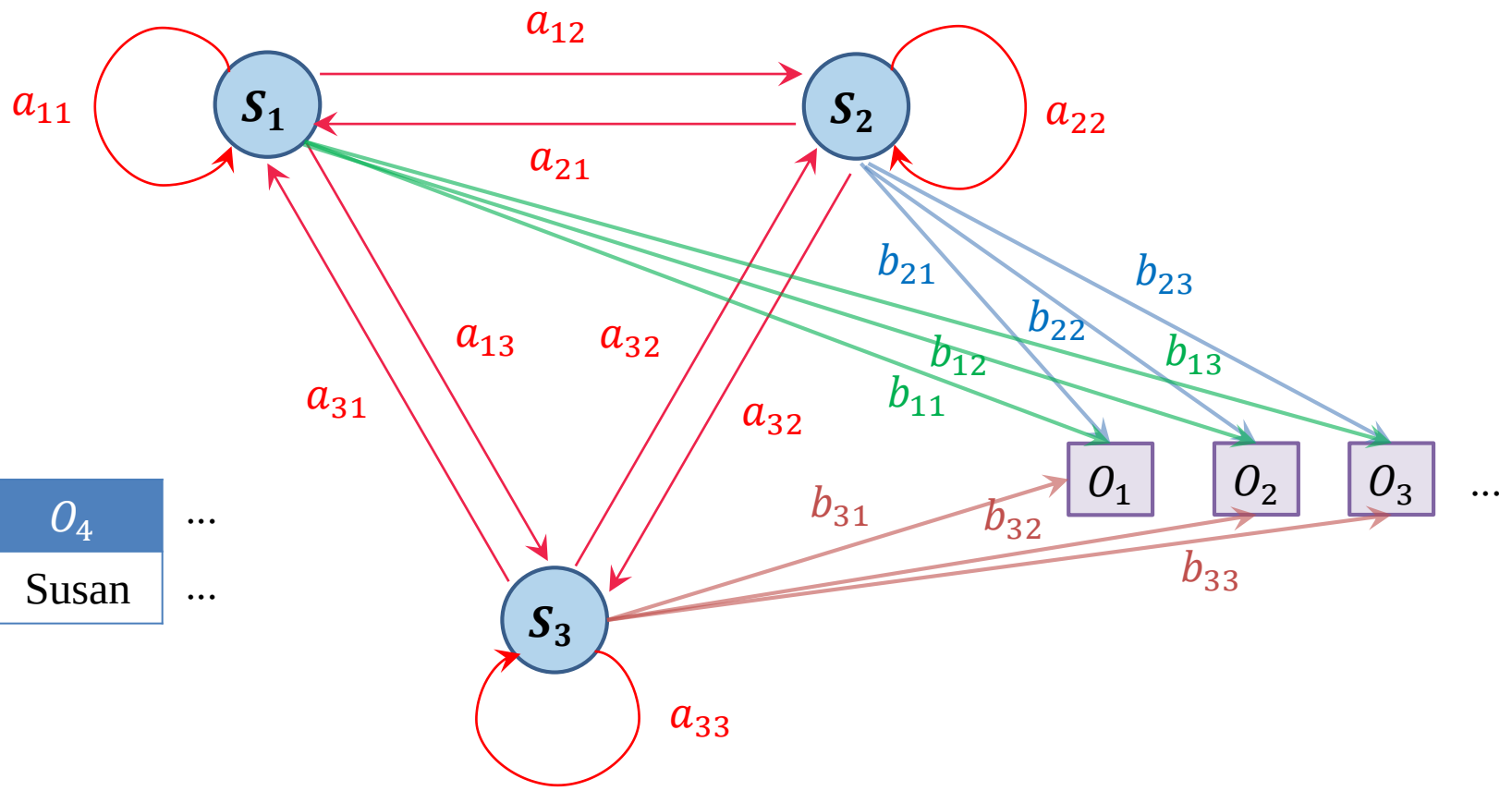


은닉 마르코프 모델 : 품사 태깅

품사 태깅 문제: 다음과 같은 은닉 마르코프 모델을 생각할 수 있다.

S_1	S_2	S_3
N	**V**	**M**

O_1	O_2	O_3	O_4	...
John	will	meet	Susan	...



은닉 마르코프 모델 : 디코딩 문제

디코딩 문제:

- 은닉 상태의 시퀀스 $x_1, x_2, x_3, \cdots, x_T$와 관찰값의 시퀀스 $y_1, y_2, y_3, \cdots, y_T$를 전제해 본다.
- 은닉 상태 x_t는 y_t를 통해서 나타난다고 생각할 수 있다.

  $\Rightarrow$ 그러므로 은닉 상태에 대한 정보는 다음 조건부 확률을 통해서 알 수 있다.

$$p(x_T, \cdots, x_2, x_1 | y_T, \cdots, y_2, y_1)$$

  $\Rightarrow$ 이것을 베이즈 정리를 적용하여 다음 비례관계로 표현할 수 있다.

$$P(x_1, x_2, \cdots, x_T | y_1, y_2, \cdots, y_T) \propto P(y_1, y_2, \cdots, y_T | x_1, x_2, \cdots, x_T)\, P(x_1, x_2, \cdots, x_T)$$

- 가장 유력한 은닉 상태 시퀀스를 찾아내고자 한다면 사후확률을 최고화 (maximize) 해야 한다.

  $\Rightarrow$ "우도" $P(y_1, y_2, \cdots, y_T | x_1, x_2, \cdots, x_T)$의 최고화를 통해서 달성.

  $\Rightarrow$ "Viterbi 알고리즘".

실습 #0103

→ 사용: **ex_0103a.ipynb , ex_0103b.ipynb** ←

순서

1. 확률 모델링:

    1.1. 언어 모형.

    1.2. 나이브 베이즈 분류기.

    1.3. HMM 모델과 활용.

    1.4. 연관성 분석 (추천 시스템).

추천시스템 : 개요

추천시스템에 대해서:

- 추천시스템은 다양한 서비스 영역에서 활용 가능하다. 예). Youtube, 쿠팡, Netflix, 등.
- 추천시스템의 유형.

a). 고객에 개별화 되지 않은 추천.

⇒ 다수의 평점 또는 "좋아요" 클릭의 평균. 이외의 다양한 랭킹 또는 Score 기반 추천.

⇒ 지도학습, **연관성 분석**, 등의 방법.

b). 고객에 개별화 된 추천.

⇒ 협업 필터링 (Collaborative Filtering).

⇒ **행렬 분해** (Matrix Factorization).

추천시스템 : Netflix Prize

Netflix Prize (2007년 ~ 2009년):

- Netflix가 주관한 영화추천 알고리즘 공모전.

  ⇒ 100만 불 (한화 11억) 상금.

  ⇒ 48만 고객과 1만 7천 영화에 대한 rating (평점) 데이터 제공.

NETFLIX

추천시스템 : Netflix Prize

Netflix Prize (2007년 ~ 2009년):

- 결과:

  ⇒ AT&T사내 팀 “BellKor”의 “Pragmatic Chaos”가 최종 우승함.

  ⇒ 협업 필터링 기반의 추천 알고리즘.

  ⇒ 기존의 예측오류를 10%이상 줄임: RMSE ~ 0.8567.

연관성 분석 : 개요

연관성 분석 (Association Analysis)에 대해서:

- **개별화 되지 않은** 추천 시스템 유형에 해당한다.
- 일명 “장바구니 분석” 이라고도 불리운다.
- 여러 번 발생한 이벤트나 거래에서 일정한 규칙을 찾아내는 분석이다.
- 마케팅, 바이오 인포매틱스, 질병진단 등의 목적으로 많이 활용된다.

연관성 분석 : 개요

연관 규칙 (Association Rule):

- A와 B는 독립적으로 발생할 수 있는 사건이다.

  예). A=맥주를 구매한다, B=기저귀를 구매한다.

- 연관 규칙은 다음과 같이 표기한다.

$$A \rightarrow B$$

연관성 분석 : 지지도

연관 규칙 (Association Rule): 지지도 (Support).

- 전체 사건 공간에서 A와 B가 동시에 발생하는 비중이다.
- 해당 규칙이 얼마나 의미 있는 규칙인지 나타낸다.

$$Supp(A \rightarrow B) = P(A \cap B)$$

연관성 분석 : 신뢰도

연관 규칙 (Association Rule): 신뢰도 (Confidence).

- A를 전제한 후 B가 발생할 확률이 얼마나 높은지를 나타낸다.
- A와 B가 동시에 발생하는 확률과 A가 독립적으로 발생하는 확률 사이의 비율과도 같다.

$$Conf(A \rightarrow B) = P(B|A) = \frac{P(A \cap B)}{P(A)}$$ ⇐ "확률의 곱셈 법칙"

연관성 분석 : 향상도

연관 규칙 (Association Rule): 향상도 (Lift).

- A를 전제한 후 B가 발생할 확률과 B가 완전히 우연으로 발생할 확률 사이의 비율이다.
- A와 B 사이의 “상호 관계”의 강도를 나타내어 준다.

$$Lift(A \rightarrow B) = \frac{P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(A)P(B)}$$

- 클수록 강한 연관 규칙을 의미한다 (최소 1 이상).

연관성 분석 : 알고리즘

연관 규칙 (Association Rule): 알고리즘.

- 수많은 규칙을 모두 하나씩 고려하지 않고 다음과 같이 강한 규칙만을 걸러낼 수 있다.
- Apriori 알고리즘.

  ⇒ 2000년도 전에 개발된 1세대 알고리즘이다.

  ⇒ 최소 지지도 이상의 빈발 (frequent) 집합 대상으로만 연관 규칙을 계산해 준다.
- FP-Growth 알고리즘.

  ⇒ 비교적 최근에 개발된 알고리즘이다.

  ⇒ FP-Tree를 만들어서 손쉽게 최소 지지도 이상의 규칙을 만들 수 있는 방법이다.

실습 #0104

→ 사용: **ex_0104a.ipynb , ex_0104b.ipynb** ←

모듈 #1 : 끝

문의:

sychang1@gmail.com